
CST1204: Introduction to Databases

Jun Shan
Adjunct Professor
City Tech, CUNY

Week 10 Session 2

10/30/2019

Agenda

- GROUP BY deep dive
- Data flow and SQL review

SELECT Execution Order

1. Going through row by row first: SELECT columns with WHERE
2. If GROUP BY is specified
 - a. Aggregate
 - b. Apply HAVING
3. Apply alias
4. Apply ORDER BY

SELECT Column Availability

- Using columns with GROUP BY
 - Non-aggregated column(s) in SELECT must be in GROUP BY
 - Column(s) in HAVING or ORDER BY must be in SELECT
- Using columns without GROUP BY
 - Column(s) in ORDER BY may not be in SELECT
- Using alias in SELECT
 - Can not use alias in WHERE, HAVING, or GROUP BY
 - Can use alias in ORDER BY

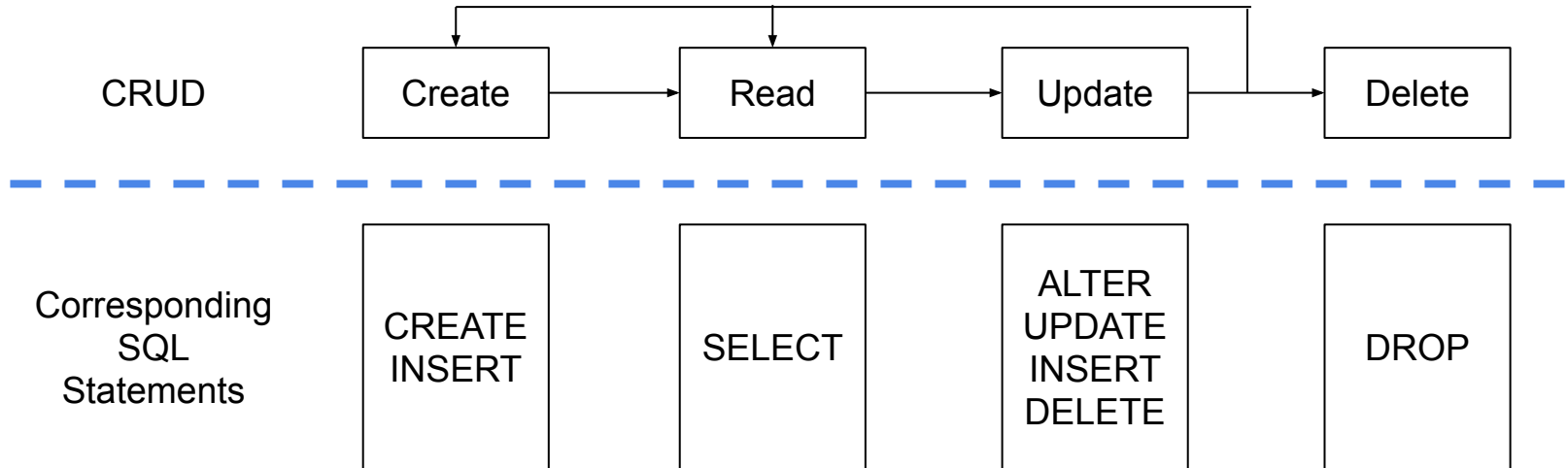
GROUP BY

```
SELECT <group by columns>, <aggregated columns>  
FROM <table>  
WHERE <boolean expression of table columns>  
GROUP BY <group by columns>  
HAVING <boolean expression of group by columns>  
ORDER BY <subset of group by columns>
```

Data Lifecycle

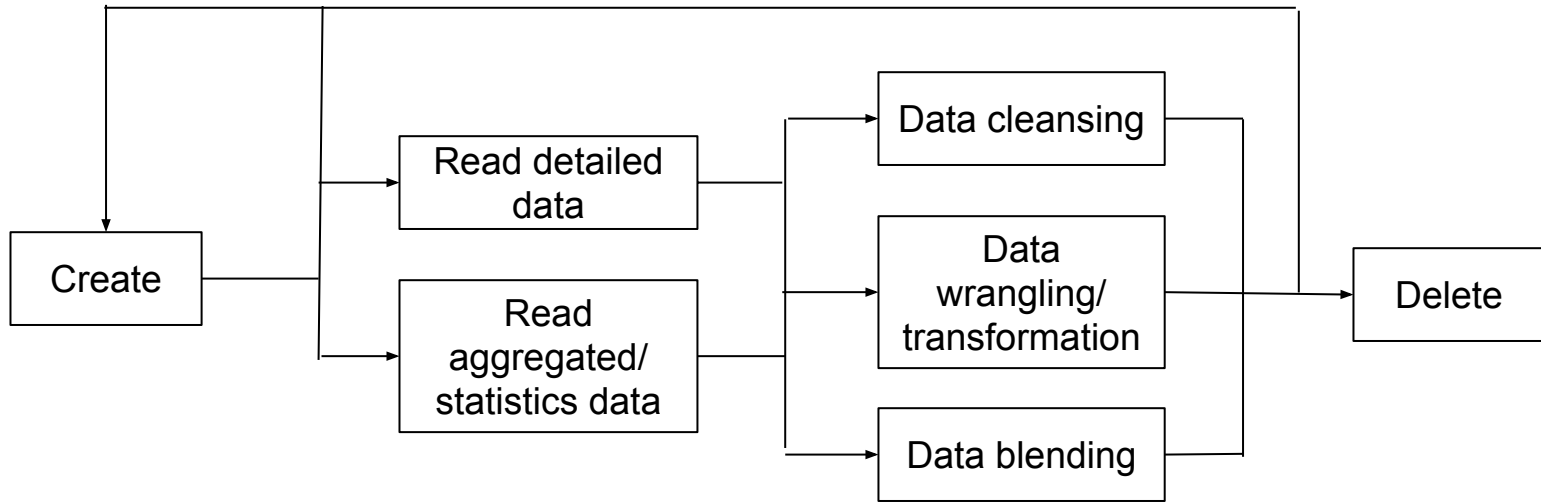
How data exists in computer systems: Create, Read, Update, Delete (CRUD)

- Generic lifecycle of data in ANY computer system, not SQL specific.



Data Lifecycle Operations

Read and Update can be further categorized

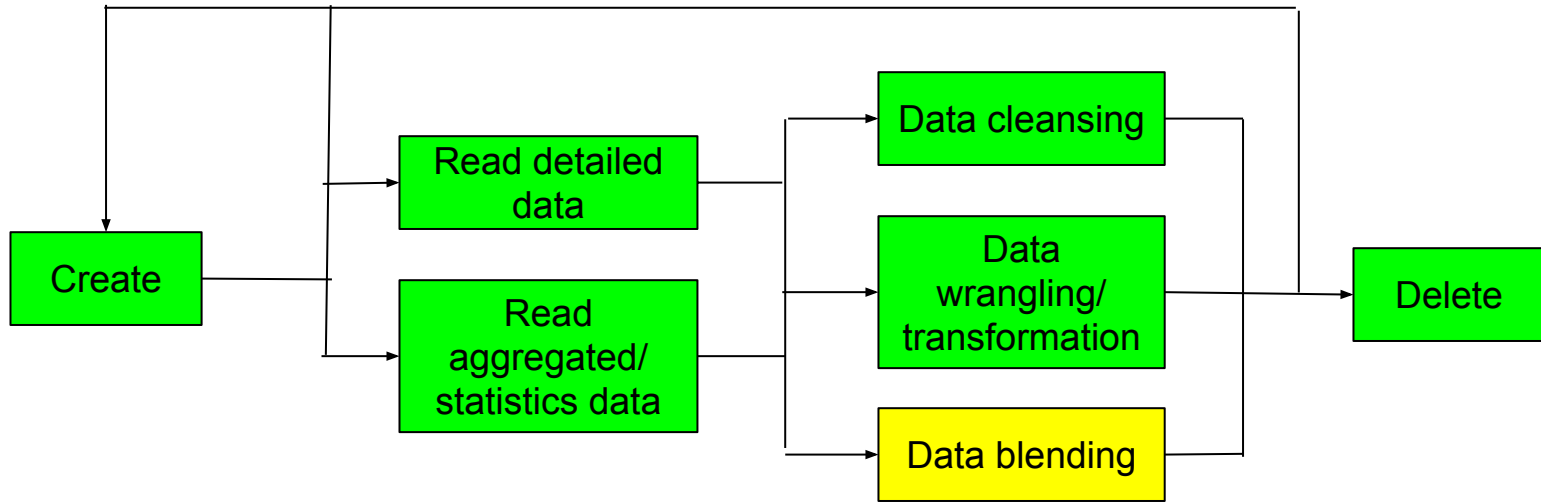


Data Lifecycle Operations in SQL

Data Lifecycle Operations in SQL:

- Read detailed data: SELECT
- Data cleansing: UPDATE, DELETE, INSERT
- Data wrangling/transformation: INSERT, SELECT AS, Functions
- Read aggregated/statistics data: SELECT ... GROUP BY
- Data blending: JOIN, UNION

Data Lifecycle Operations



TEST THREE on 11/06

- Same pattern for TEST TWO, THREE, FOUR, and FINAL
 - CREATE based on entity/relationship description
 - INSERT INTO
 - Update value in a column in the given row: regular value or NULL
 - Add a new column or modify an existing column
 - All sorts of SELECT ... WHERE ... GROUP BY ... HAVING
 - DROP this table

CREATE and DROP

- CREATE
 - Based on shorthand representation
 - Data type: INT, CHAR, VARCHAR, DATE
 - Primary Key definition
 - Multi-column primary key
- DROP
 - DROP TABLE <table>: No column name or * or FROM
 - Drop one table at a time. No multi=drop.

ALTER

- ALTER syntax

```
ALTER TABLE <table> ADD/MODIFY ...
```

INSERT

- Syntax

```
INSERT INTO <table> VALUES ()
```

- Remember to add single quote for CHAR/VARCHAR

- INSERT from existing table

```
INSERT INTO <new_table>
```

```
SELECT ... FROM <existing_table>
```

- INSERT NULL

UPDATE

- Syntax

```
UPDATE <table> SET <col> = <value> WHERE ...
```

- Update scenarios

- 10% increase/decrease
- UPDATE column to NULL
- UPDATE rows with NULL values

DELETE

- Syntax

`DELETE FROM <table> WHERE`

- No column names because it removes the entire rows

- DELETE vs DROP vs ALTER TABLE ... DROP COLUMN

- DROP removes the table definition
- DELETE removes data from table but table is still there
- ALTER TABLE ... DROP COLUMN removes column from table

Single Table SELECT

- Syntax

```
SELECT <column_list> FROM <table> WHERE ...
```

- SELECT scenarios

- Select all rows
- Select all columns
- Select column/computation and give it a new name (alias)
- Select values but only display it once

WHERE Clause

- Simple condition and computed condition
- Multiple conditions combined with AND/OR/NOT
- NULL handling: `IS NULL` or `IS NOT NULL`
- IN: using OR to replace IN
- BETWEEN ... AND
 - Using `>=` AND `<=` to replace BETWEEN AND
 - Lower bound must always be lower than upper bound

WHERE for CHAR/VARCHAR

- Case sensitive
 - Lower case is always greater than upper case. Upper case is always greater than numbers.
 - How to find all 'city', regardless of case
- LIKE
 - Find rows that a column starts/ends with 'C'
 - Find rows that a column contains 'city'

NULL in WHERE Clause

- NULL handling:
 - `IS NULL` or `IS NOT NULL`
 - Any expression involving NULL will return NULL. Neither TRUE nor FALSE

ORDER BY

- Sorting order: ASC (default) or DESC
- Scenario
 - Order Item within Warehouse
 - Order by Warehouse first, then Item

GROUP BY

- SELECT ... GROUP BY
 - Non-aggregated columns must show up in both SELECT and GROUP BY
- HAVING
 - Used only on aggregated data set
 - Compare with WHERE

Hands On

- Chapter 4 exercise questions for Colonial Adventures 7-20

Homework

- Chapter 4 exercise questions for Colonial Adventures 7-20
- Chapter 8 exercise questions for Solmaris Condominium Group 1-3